NASA Technical Memorandum 113165

Design Master: A System Engineering Graphical User Interface Tool

D.W. Worthem
NYMA, Inc.
Brook Park, Ohio

L.V. Truong
Lewis Research Center
Cleveland, Ohio

G. Kusic
University of Pittsburgh
Pittsburgh, Pennsylvania

Prepared for
Technology 2007
sponsored by NASA Tech Briefs Magazine, National Aeronautics and
Space Administration, The Federal Lab Consortium, and
The Tech Utilization Foundation
Boston, Massachusetts, September 22–24, 1997

National Aeronautics and
Space Administration

DESIGN MASTER:
A SYSTEMS ENGINEERING GRAPHICAL USER INTERFACE TOOL

D.W. Worthem, Software Engineer
Nyma, Inc., 2001 Aerospace Parkway, Brookpark, OH 44142
Dennis.W.Worthem@lerc.nasa.gov, ph: 216-977-1041, fax: 216-977-1269

L.V. Truong, Electrical Engineer
NASA Lewis Research Center
Cleveland, OH 44135

G. Kusic, Professor of Electrical Engineering
University of Pittsburgh
Pittsburgh, PA 15261

ABSTRACT

Design Master is a desktop personal computer program for use by system designers that allows simulation type tradeoff design studies for various engineering systems. The program is a generic dynamic object-oriented program with a graphical event-driven user interface. The program is fully integrated into the Windows 95 and Windows NT operating system environments and provides connectivity across the network and with other programs through the COM (Component Object Model). Design Master permits the creation of templated objects, groups of objects, or entire system representations for particular engineering disciplines.

INTRODUCTION

Software modeling or simulation tools for engineering applications often exist as separate packages (power, scheduling, mass, cost, reliability, etc.). This does not allow system designers to explore the synergies among various aspects of the "total" system. Because of the lack of commercial tools with the ability and ease to integrate engineering models and data bases, we designed a new and reusable tool for a wide range of engineering applications. It is called Design Master: Systems Engineering Graphical User Interface Tool. This paper introduces Design Master and discusses some of its typical engineering applications.

DESCRIPTION

Figure 1 shows the basic object hierarchy of the program. On the left side are the underlying software objects that represent the real world objects that the user wishes to simulate. On the right side are the visual objects used to view, edit, and manipulate the visual representation of the underlying software objects on the computer screen. The software objects on the left have a collection of system objects parented by the application. Each system object is the top level object in any one loaded system. Under each system is a collection of subsystems, each of which is an object parented by the system. Each subsystem in turn has a collection of subsystems, therefore providing any level of embedding in the hierarchy. The name of a system or subsystem is provided by the user.

Under each system or subsystem are objects that contain properties pertaining to that object. System or Subsystem Properties contain various object properties such as connection information and any executable code modules embedded at the level in the hierarchy. View Properties contains the properties of the child window representing that object, such as background color. Button properties contains the properties of an object's visual representation within its parent's child window. The system object has Default Button Properties instead because it is a top level object and is not contained within another child window. The default values are for the primitive visual objects used throughout the system. Custom properties are the user defined properties for the real world object represented.

The separation of software objects, representing a real world system from the software objects for visual objects on the computer screen, allows for a more intuitive object hierarchy and permits unloading selected child windows to save computer system resources. The visual properties for a system or subsystem are subclassed under that system or subsystem and can be readily accessed to create the visual objects for that system or subsystem. Relationships between visual objects are established by the system hierarchy rather than establishing links between visual objects.

Figure 2 shows a screen capture of the application's main window with a system loaded that contains multiple levels of subsystems. The system is a space station electrical power system with three power modules. At the upper left is the child window showing the top level system. Each object representing a module is a container of objects constituting the module. Pointing and clicking on a module object brings up the child window representing the module and showing the child objects within the module. The three module windows are the other two windows on the left side and the lower right side. The other child window shows what is contained within an object parented by the first module. This child window demonstrates a balloon that pops up when the mouse cursor is moved over an object. The balloon shows a textual identification of an object and assists in precise targeting when drawing a connector from one object to another or identifying objects on a crowded window. Also shown is a context menu that pops up when the mouse right button is clicked on an object (a similar context menu appears for the background of the child window).

The caption in the title bar of each child window shows the object path in the hierarchy of the object represented by the child window. The main window caption shows the caption of the currently active child window. Connecting lines can be connected either directly on to an object or can be connected to a port object (green in Figure 2) on the target object. User-defined property values are optionally passed from object to object through a port connection. Figure 2 also shows application toolbars docked at the top of the main window. The floating toolbar shown in the upper left area of the main window is a user-defined toolbar that contains template objects available for reuse. Clicking on a template toolbar button permits a template object to be pasted on any of the child windows. The template toolbars can be saved to file and reopened for reuse in another system or provided to another user.

The visual objects can also serve as object labels and textual or graphical annotations. Calculated or real time values of properties can be displayed in any cell on a visual object.

Figures 3 and 4 are examples of the extensive dialog boxes and property sheet provided to customize the system objects and their visual representation. Figure 3 shows 3 of the 8 property pages used to set the properties of the buttons (visual objects on a child window). Design Master provides extensive capabilities to design complex shapes, provide multiple cells on the button, each with separate text, bitmaps and/or colors. Most of the dialog boxes and property pages are modeless so that a series of property changes can be made for evaluation without having to reopen them each time. The program also has a visual designer tool for mouse point, click, and drag design of objects.

The top of Figure 4 show the dialog box that lists the user-defined properties for an object. These properties would be physical or engineering properties or properties to manage or describe the object within the context of an entire system. User-defined property values can be strings, short or long integer, single or double floating point values, booleans and multidimensional arrays of any of these types. Through the use of a Visual Basic for Applications compatible scripting tool, object properties can be accessed and exported to an external system solver or the solver program can be created within Design Master. The access is through COM Automation. Calculated values then can be imported back into the object's set of stored properties. The scripting tool is shown on the lower left of Figure 4. The scripter provides a Integrated Development Environment (IDE) complete with runtime debugging capability and an object browser to ascertain the methods and properties of an object and any parameters of the methods. The IDE also has a dialog box editor to create custom interface elements. The scripter allows the Design Master to be a client of any automation server. The program as a whole is an automation server.

The following table lists most of the main features of the program:

Design Master's Features
Create any number of subsystem levels or objects within a level up to the limit of computer resources.
Represents each subsystem as object on parent window or as a containing window.
Multiple cells in visual object, each containing text, bitmaps, and/or colors.
Multiple segments for customized shape of visual objects.
Floating popup balloon provides textual identification of visual objects. Balloon properties can be edited.
Visual Designer for visual objects provides interface for mouse point, click, and drag design of objects.
Property pages provide for extensive editing of visual object properties.

Property pages provide for extensive editing of visual container (window) properties.
Visual Basic for Applications compatible scripting tool with runtime debugger capabilities, object browser, and a custom dialog box editor to create a customized interface. Scripter also optionally responds to events in the program providing a customized response to events such as clicking the mouse on an object.
Port objects for connecting visual objects with connectors. Optionally fixes connection points.
Single and polyline segment connectors (number of segments can be changed) with or without arrowheads.
Treats connectors as objects with custom defined properties.
User creates and defines properties through dialog boxes.
User optionally defines property values as strings, short or long integer, single or double floating point values, booleans and multidimensional arrays of any of these types.
Stitches property values optionally between connected objects.
Exposes system, subsytem and their properties through COM-based Automation.
Program is a COM-based Automation server and is an Automation client through the scripter.
Graph object provides basic graphing capabilities for object properties and calculated values.
Embeds object help documentation within a system data file that also contains Automation embedded or linked items. Edit object help documentation by a special version of WordPad.
Underlying object hierarchy separates the abstract object from the visual representation of the object.
System or subsystem contains an optional object reference to any other arbitrary system or subsystem that is loaded.
Embeds scripter code modules in an object executed from multiple locations in the user interface. User optionally embeds code modules within each other and/or defines an execution order for the module.
File structure uses the COM Structured Storage specification that is object-oriented and has security features. File structure viewer utility provides a tree view of a file with direct access to any level of the hierarchy.
Tree view shows hierarchy of all loaded systems and provides direct access to properties at any level in the hierarchy.
Status bar shows load and unload progress and coordinates of move and resize operations.
Move and resize objects singly or in groups and maintain aspect ratio during resizing.
Zoom views in or out, create multiple copies of views, split and scroll views manual or automatically.
Save objects to floating toolbars to be used as templates. Save the template toolbars to file and reload and create new systems. The template toolbar buttons contain user-created bitmaps and user-defined text for tool tips to identify the contents of the button. Edit the contents of any template toolbar button.
Extensive main menus, toolbars and context menus provide easy access to properties and traversing the hierarchy.
Drag and drop of properties.
Copy and paste of objects and groups of objects to other views and locations within a view.
Horizontal and vertical rulers for each view.
Each view has an optional grid and objects snapped to grid.
Full printing support for individual or multiple views.
Interface customization such as allowing user-selected definition of mouse click operation, automatic saving of files, placement and size of windows, multiple instances of dialog boxes, etc.
Window and dialog box title bars provide the object path in the hierarchy to the referenced object.
Undo and redo operations.
Startup directory for automatic loading of system and template toolbar files.

SUMMARY

Design Master provides a flexible, extensible, and low-cost environment for system engineering simulation. The program provides a very broad capability to visually design a system, customize the environment, define properties, write and test code to access properties and/or perform calculations on the properties. As a COM Automation enabled program, it can be used effectively with other programs that the user may have on their desktop computer or residing on other computers on the network using the Distributed Component Object Model (DCOM). The Visual Basic for Applications compatible scripting language is a human-readable object-based language that is easy for a system designer to learn and use. Evaluation copies of the program are available by contacting the authors.

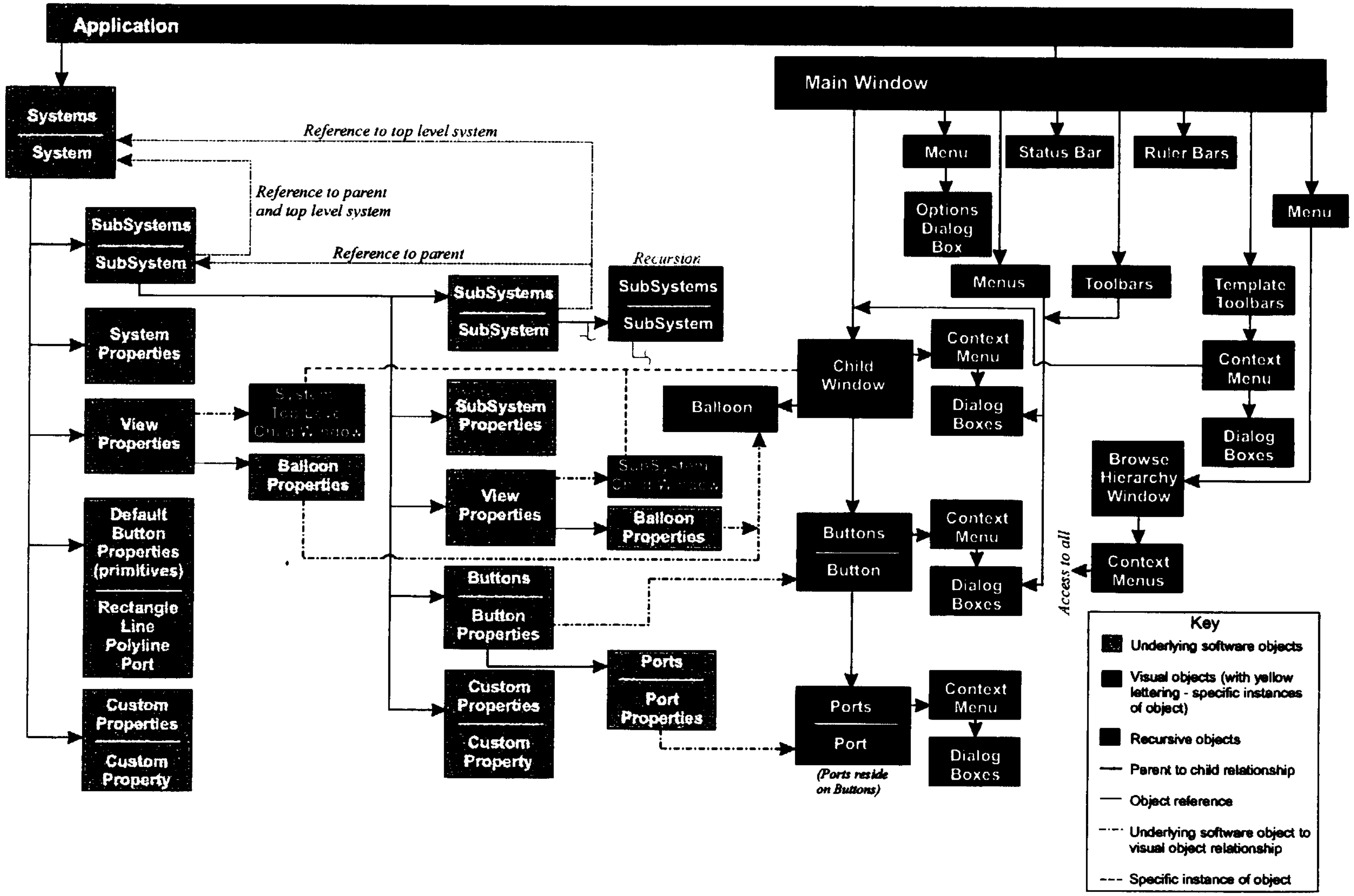


Figure 1. - Program object hierarchy consists of objects representing real world objects (on the left in gray) and their visual representation in the graphical user interface (on the right in red).

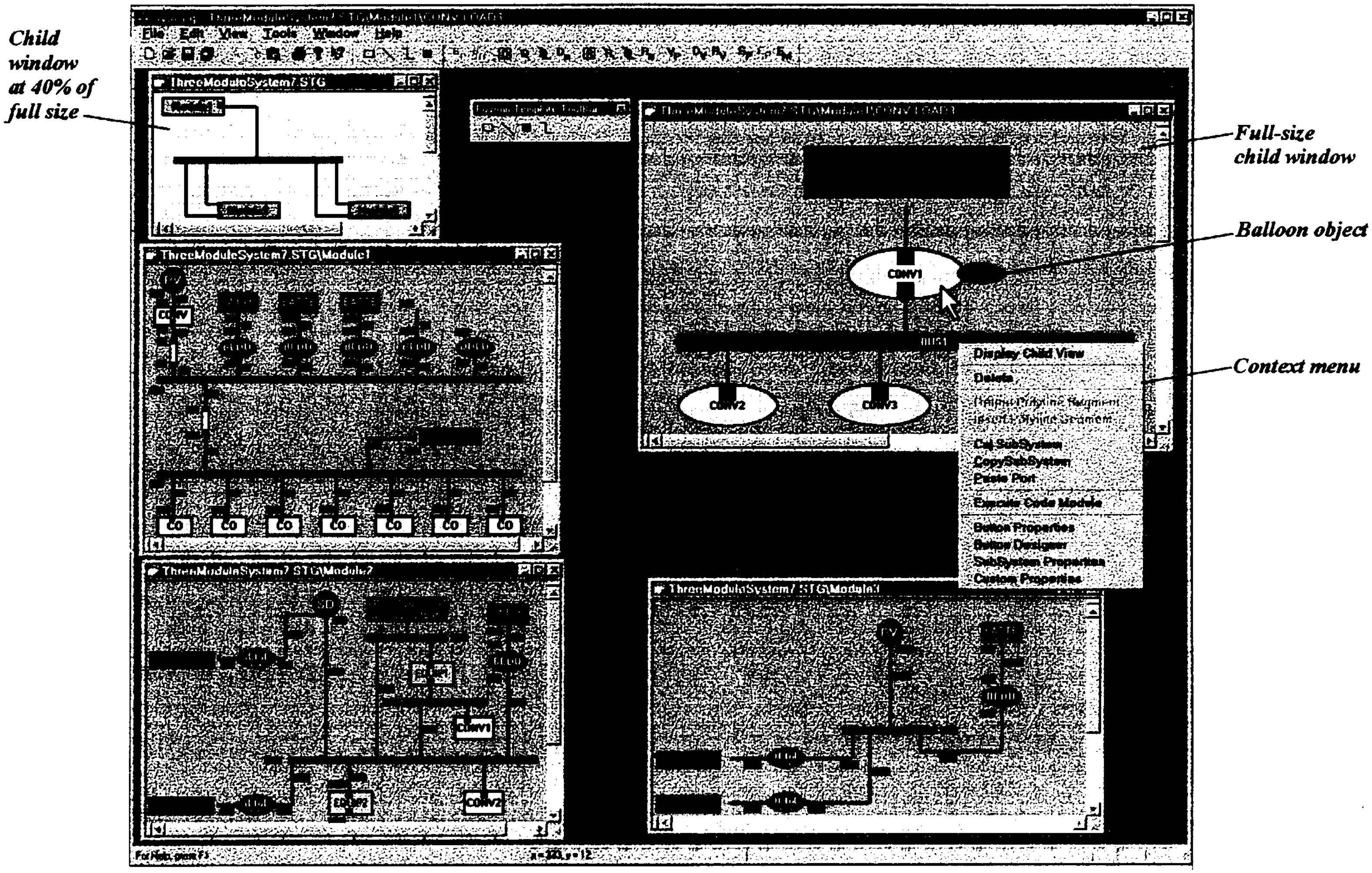
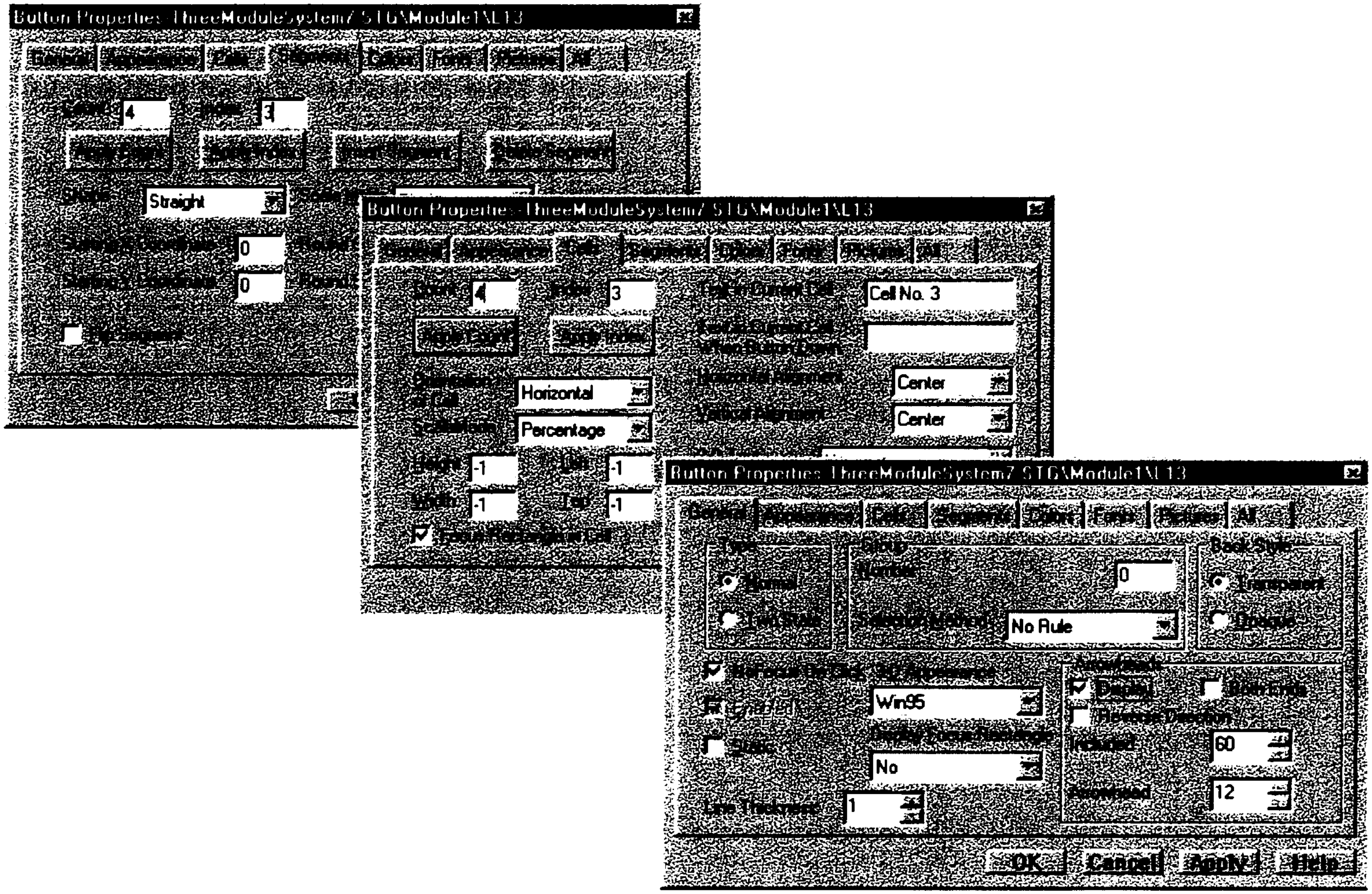


Figure 2. - Main window shows the various hierarchy levels in a three power module space station power system using child windows. A balloon pops up to provide textual identification of an object. Right clicking on an object provides a context menu with access to an extensive customization capability. Template toolbar provides for customized reusable objects.



Button Properties - ThreeModuleSystem7 STG\Module1\L13

Count: 4 Index: 3

Shape: Straight

0

0

Button Properties - ThreeModuleSystem7 STG\Module1\L13

Count: 4 Index: 3

Cell No. 3

Horizontal

Percentage

Center

Center

Height: -1 Left: -1

Width: -1 Top: -1

☑ Focus Rectangle in Cell

Button Properties - ThreeModuleSystem7 STG\Module1\L13

Type: ● Normal ○ Two State

Group Number: 0

No Rule

Back Style: ● Transparent ○ Opaque

☑ 3D Appearance: Win95

☑ Enabled

☐ Static

No

Line Thickness: 1

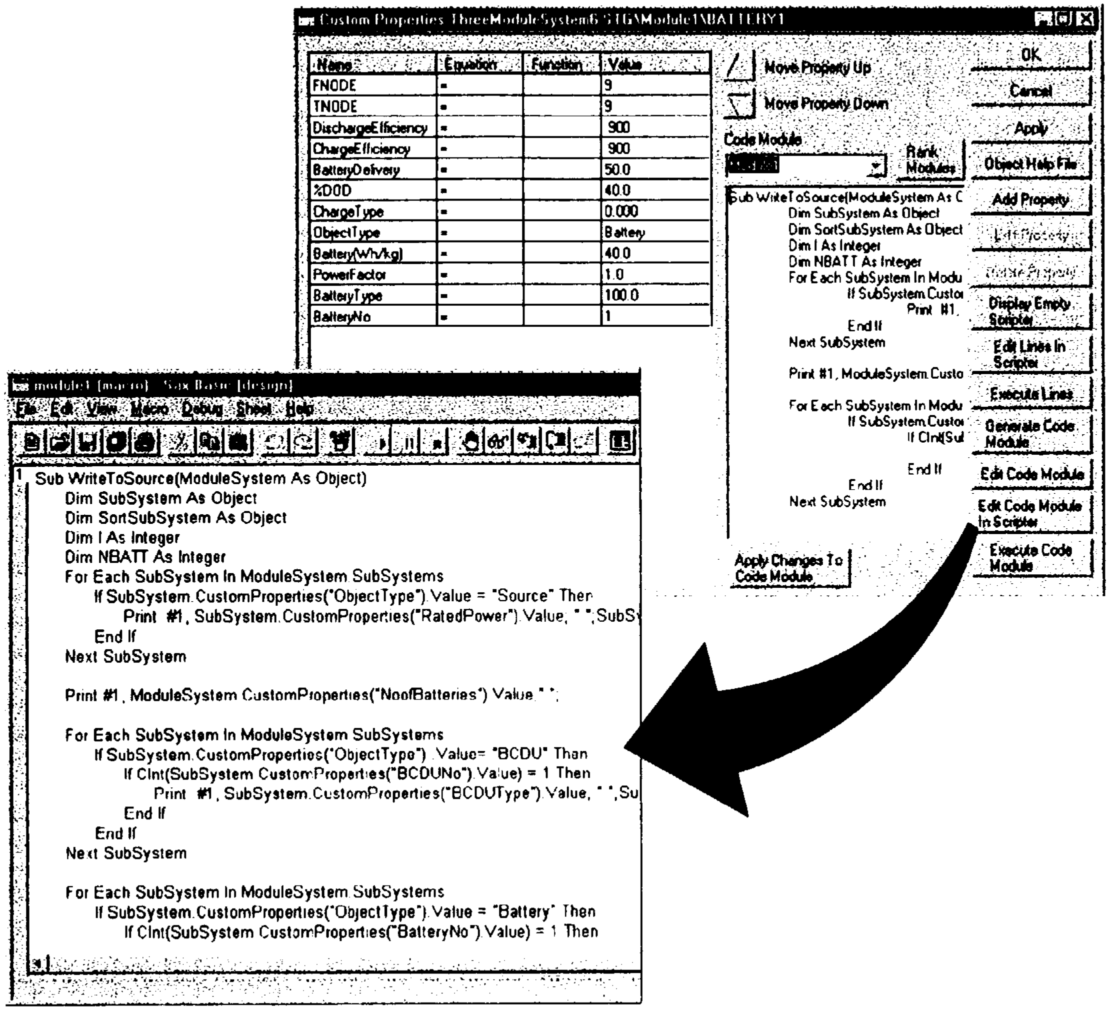
☑ Display ☐ Both Ends

☐ Reverse Direction

60

12

Figure 3. - Three of the eight property pages used to set object visual properties demonstrate the extensive customization capabilities of the program.



Custom Properties ThreeModuleSystem6 STG\Module1\BATTERY1

Name	Equation	Function	Value
FNODE	=		9
TNODE	=		9
DischargeEfficiency	=		900
ChargeEfficiency	=		900
BatteryDelivery	=		50.0
%DOD	=		40.0
ChargeType	=		0.000
ObjectType	=		Battery
Battery(Wh/kg)	=		40.0
PowerFactor	=		1.0
BatteryType	=		100.0
BatteryNo	=		1

```
Sub WriteToSource(ModuleSystem As C
        Dim SubSystem As Object
        Dim SortSubSystem As Object
        Dim I As Integer
        Dim NBATT As Integer
        For Each SubSystem In Modu
                If SubSystem.Custo
                        Print  #1,
                End If
        Next SubSystem

        Print #1, ModuleSystem.Custo

        For Each SubSystem In Modu
                If SubSystem.Custo
                        If CInt(Sul

                        End If
                End If
        Next SubSystem
```

module1 (macro) - Sax Basic [design]

```
Sub WriteToSource(ModuleSystem As Object)
    Dim SubSystem As Object
    Dim SortSubSystem As Object
    Dim I As Integer
    Dim NBATT As Integer
    For Each SubSystem In ModuleSystem SubSystems
        If SubSystem.CustomProperties("ObjectType").Value = "Source" Ther
            Print  #1, SubSystem.CustomProperties("RatedPower").Value, " ";SubS
        End If
    Next SubSystem

    Print #1 , ModuleSystem CustomProperties("NoofBatteries") Value " ";

    For Each SubSystem In ModuleSystem SubSystems
        If SubSystem.CustomPropertios("ObjectTypo") .Valuo= "BCDU" Than
            If CInt(SubSystem CustomProperties("BCDUNo").Value) = 1 Then
                Print  #1, SubSystem.CustomProperties("BCDUType").Value, " ",Su
            End If
        End If
    Next SubSystem

    For Each SubSystem In ModuleSystem SubSystems
        If SubSystem.CustomProperties("ObjectType").Value = "Battery" Then
            If CInt(SubSystem CustomProperties("BatteryNo").Value) = 1 Then
```

Figure 4. - User defines custom properties in a series of dialog boxes (top). Properties can be accessed and used in executable code modules by a Visual Basic for Applications compatible scripting language Integrated Development Environment (bottom).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE
September 1997

3. REPORT TYPE AND DATES COVERED
Technical Memorandum

4. TITLE AND SUBTITLE
Design Master: A Systems Engineering Graphical User Interface Tool

5. FUNDING NUMBERS
WU-297-30-00-00

6. AUTHOR(S)
D.W. Worthem, L.V. Truong, and G. Kusic

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

8. PERFORMING ORGANIZATION REPORT NUMBER
E-10924

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA TM-113165

11. SUPPLEMENTARY NOTES
Prepared for Technology 2007 sponsored by NASA Tech Briefs Magazine, National Aeronautics and Space Administration, The Federal Lab Consortium, and The Tech Utilization Foundation, Boston, Massachusetts, September 22-24, 1997. D.W. Worthem, NYMA, Inc., 2001 Aerospace Parkway, Brook Park, Ohio 44142 (work funded by NASA Contract NAS3-27186); L.V. Truong, NASA Lewis Research Center; and G. Kusic, University of Pittsburgh, Pittsburgh, Pennsylvania 15261. Responsible person, L.V. Truong, organization code 5490, (216) 433-6153.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 61

This publication is available from the NASA Center for AeroSpace Information, (301) 621-0390.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Often software modeling or simulation tools for engineering applications exist as separate packages (power, scheduling, mass, cost, reliability, etc.). This does not allow system designers to explore the synergies among various aspects of the "total" system. Because the lack of commercial tools with the ability and ease to integrate engineering models and databases, we designed a new and reusable tool for a wide range of engineering applications. It is called the Design Master: A Systems Engineering Graphical User Interface Tool. Design Master is a desktop personal computer program for use by system designers which allows system simulation or design tradeoff studies for various engineering applications (electrical, thermal, reliability, etc.). The program is generic and dynamic object-oriented with graphical event-driven user interface. It is fully integrated into Microsoft Windows environments and provides connectivity across network and other executable programs through the Component Object Model (COM) feature. Design Master also permits the creation of objects (graphical screen images with design functionalities or properties), template objects, groups of objects, or entire system representations for a particular engineering discipline. New models and databases can be created within the system. However, existing shareware models and databases can be linked to the Design Master for execution. This paper introduces the Design Master and discusses some of its typical engineering applications.

14. SUBJECT TERMS
Graphical user Interface; Engineering software tools; Object oriented simulation; Design and analysis software tools

15. NUMBER OF PAGES
09

16. PRICE CODE
A02

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT